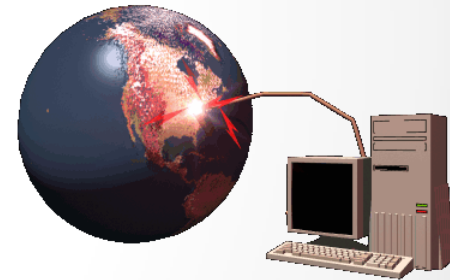




Teknik Informatika S1

SOFTWARE QUALITY AND TESTING

Perencanaan Pengujian



Disusun Oleh:

Egia Rosi Subhiyakto, M.Kom, M.CS

Teknik Informatika UDINUS

egia@dsn.dinus.ac.id

+6285640392988

SILABUS MATA KULIAH

8. Perencanaan Pengujian + Tugas Black Box

9. Presentasi Tugas Black Box

10. White Box (1)

11. White Box (2) + Post Test

12. Manajemen Fungsi Testing

13. OO Testing

14. Post Test/ Presentasi Tugas White Box

Mengapa Perencanaan Testing?

1. Karena pelanggan biasanya hanya memiliki sedikit kesabaran terhadap produk yang tidak memenuhi kualitas yang mereka harapkan.
2. Tanpa adanya perencanaan dan organisasi, cakupan dan realibilitas dari pemenuhan usaha tes hanyalah berupa dugaan
3. Tanpa adanya perencanaan dan organisasi, estimasi kebutuhan jadwal dan sumber daya tes, penilaian kesiapan sistem untuk diserahkan berupa coba-coba dalam suatu kondisi yang penuh dengan ketidakpastian.

Mengapa Perencanaan Testing?

4. Sistem modern, dengan teknologi GUI, *Client/ Server*, dan teknologi baru lainnya, adalah sangat komplek, dan banyak produk atau subsistem yang membutuhkan untuk diintegrasikan dan bekerja bersama.
5. Serta tanpa organisasi yang efektif, efisiensi testing adalah rendah.

Mengapa Perencanaan Testing?

- ✓ Suatu rencana tes mendeskripsikan:
 - Aktivitas testing
 - Komponen-komponen yang dites
 - Pendekatan testing
 - Tiap alat bantu yang dibutuhkan
 - Sumber daya dan jadwal
 - Resiko dari aktivitas testing.

Obyektivitas Rencana Testing?

Obyektivitas utama dari rencana Testing adalah:

1. Memfasilitasi tugas-tugas teknis dari testing
2. Meningkatkan komunikasi tentang tugas-tugas dan proses-proses testing
3. Menyediakan struktur untuk pengorganisasian, penjadwalan, dan pengaturan proses testing.

Tugas-tugas teknis testing

1. Meningkatkan cakupan tes: Daftar fitur, komponen, layar, pesan kesalahan, konfigurasi *hardware*, dan lain-lain, akan mengurangi kelalaian yang mengakibatkan kurangnya cakupan dari testing
2. Menghindarkan dari pengulangan yang tidak perlu
3. Menganalisa program untuk *test cases* yang baik.
4. Menyediakan struktur
5. Meningkatkan efisiensi tes
6. Cek pemenuhan

Komunikasi tentang tugas dan proses testing

1. Pemikiran strategi tes
2. Mengembangkan umpan balik terhadap batasan
3. Ukuran dari pekerjaan testing
4. Mengembangkan umpan balik terhadap kedalaman dan waktu
5. Akan lebih mudah untuk mendelegasikan dan mensupervisi testing suatu aplikasi

Struktur organisasi, penjadwalan dan proses

1. Mencapai persetujuan akan tugas-tugas tes
2. Mengidentifikasi tugas-tugas
3. Struktur: mengelompokkan tugas-tugas yang sama
4. Organisasi: Identifikasi siapa, bagaimana, dimana, kapan dan dengan sumber daya apa test dilakukan
5. Koordinasi: Mendelegasikan tugas berdasarkan rencana tes
6. Meningkatkan akuntabilitas: Tester mengerti tugas mereka

Kerangka Rencana Tes Sederhana

Secara sederhana dokumen rencana tes, terdiri dari:

- ✓ **Obyektifitas** → Berisi tujuan akhir yang akan dicapai oleh testing, dan produk testing yang diharapkan.
- ✓ **Strategi dan Pendekatan** → Berisi deskripsi lingkungan tes, dan cakupan dari testing.
- ✓ **Spesifikasi tes** → Berisi deskripsi tes, data masukan, kondisi inisial yang dibutuhkan, dan hasil yang diharapkan

Kerangka Rencana Tes Sederhana

- ✓ **Rencana Kerja dan Jadwal Tes** → Berisi tentang daftar tugas-tugas testing secara berurutan, kriteria dan rencana tes ulang, batasan waktu secara umum.
- ✓ **Kriteria pemenuhan.**
- ✓ **Sumber daya** → Berisi identifikasi tim tes, jam perorang yang dibutuhkan untuk testing, dan alat bantu tes otomatis yang digunakan (bila ada).

Spesifikasi Tes

Tingkat kedetailan dari suatu spesifikasi tes tergantung dari faktor:

1. Tingkat kekomplitan dan stabilitas spesifikasi sistem
2. Tingkat resiko internal produk dan fitur yang dites
3. Kredibilitas, kemampuan, dan pengalaman dari orang yang melakukan tes
4. *Back-up* dan pergantian sumber daya.
5. Tingkat otomatisasi
6. Ekstensi tes yang harus diulangi (misal untuk versi selanjutnya).

Berapa Banyak Tes Dinyatakan Cukup?

Faktor-faktor yang membantu untuk menentukan berapa banyak tes dinyatakan cukup, antara lain:

1. Cakupan fungsional yang diinginkan
2. Tingkat kualitas, reliabilitas atau kejelasan batasan yang dibutuhkan.
3. Jangkauan tipe tes yang dibutuhkan, misal kegunaan, performa, keamanan, kompatibilitas/konfigurasi.

Berapa Banyak Tes Dinyatakan Cukup?

4. Tingkat antisipasi kualitas yang telah ada di dalam sistem, bilamana diserahkan untuk dilakukan sistem testing.
5. Resiko dan konsekuensi dari *defects* yang tersembunyi dalam fitur-fitur atau aspek-aspek dari sistem tertentu.
6. Kemampuan untuk memenuhi standar audit yang telah ditetapkan, kriteria pemenuhan tes dan tujuan akhir kualitas sistem.
7. Hambatan usaha tes, seperti waktu dan sumber daya yang ada untuk testing, dan fisibilitas, kesulitan dan biaya testing

Berapa Banyak Tes Dinyatakan Cukup?

Pada dasarnya terdapat 3 faktor utama yang harus diseimbangkan dalam membuat rencana tes, yaitu:

- Tingkat kedetailan (seperti waktu dan sumber daya yang dibutuhkan untuk membuat dan merawat rencana tes)
- Tingkat organisasi dan kendali tes yang dibutuhkan.
- Kebutuhan tester dalam pengarahan tugas, otonomi dan kreatifitas.

Teknik Estimasi Usaha Tes

Barry Boehm mengidentifikasikan teknik estimasi pada proyek testing:

- ✓ Bottom-Up atau Micro-Estimating
- ✓ Top-Down atau Global-Estimating
- ✓ Formulae atau Models
- ✓ Parkinson's Law
- ✓ Pricing to Win
- ✓ Cost Averaging
- ✓ Consensus of Experts
- ✓ SWAG (Scientific Wild-Ass Guess)
- ✓ Re-Estimating by Phase

Bottom-Up atau Micro-Estimating

→ Mengembangkan rencana kerja tes detail, dan membuat estimasi waktu dan sumber daya untuk tiap tugas terkecil dari rencana kerja.

Top-Down atau Global-Estimating

→ Estimasi dimulai dari gambaran besar, dengan membandingkan cakupan dan usaha keseluruhan tes dengan usaha-usaha lain yang mirip dan menetapkan waktu sumber daya yang dibutuhkan.

Formulae atau Models

- Teknik ini menggunakan suatu formulasi untuk estimasi.
- Karakteristik yang diukur, tergantung pada formula, dapat berupa jumlah *window*, *query* atau *table* yang dites, efisiensi lingkungan tes dan jumlah *defect*. Dimana masukan karakteristik ini akan sulit untuk dibuat.
- Pembuat formula atau model tergantung pada pengalaman terhadap sistem yang akan dimodelkan.

Parkinson's Law

- Estimasi tidak hanya berupa proses kalkulasi kuantitatif, kadang faktor manusia harus dimasukan, seperti kemampuan negosiasi.
- Pendekatan ini dilakukan dengan menetapkan estimasi terhadap nilai psikologis maksimum yang dapat diterima oleh pihak bersangkutan (misal klien/ atasan).

Pricing to Win

- Pendekatan ini merupakan lawan dari Parkinson's Law, dengan menetapkan nilai estimasi terhadap nilai yang terendah.
- Teknik Parkinson's Law biasa digunakan untuk negosiasi dengan pihak luar (misal: klien), sedangkan Pricing to win digunakan untuk memberikan nilai estimasi ke dalam (misal: programmer)

Identifikasi dan Rencana Pengujian

Kelas Uji	Butir Uji	Identifikasi		Jenis Pengujian	Teknik Pengujian	Jadwal
		SKPL	PDHUPL			
Penambahan data induk mahasiswa	Penambahan dengan data baru	SKPL-11	PDHUPL11	sistem	<i>black box</i>	05/01/2004 Kumkum
	Penambahan dengan data yang sudah direkam	SKPL-11	PDHUPL12	sistem	<i>black box</i>	05/01/2004 Kumkum
Penambahan data induk kuliah	Penambahan dengan data baru	SKPL-12	PDHUPL21	sistem	<i>black box</i>	05/01/2004 Kumkum
	Penambahan dengan data yang sudah direkam	SKPL-12	PDHUPL22	sistem	<i>black box</i>	05/01/2004 Kumkum
Penambahan data hasil studi	Penambahan dengan data baru	SKPL-13	PDHUPL31	sistem	<i>black box</i>	05/01/2004 Kumkum
	Penambahan dengan data yang sudah direkam	SKPL-13	PDHUPL32	sistem	<i>black box</i>	05/01/2004 Kumkum
Login ke sistem	Data <i>userid</i> dan <i>password</i> benar	SKPL-21	PDHUPL41	sistem	<i>black box</i>	06/01/2004 Markum
	Data <i>userid</i> dan <i>password</i> salah	SKPL-21	PDHUPL42	sistem	<i>black box</i>	06/01/2004 Markum

Deskripsi Hasil Uji

Identifikasi	PDHUP11		
Nama Butir Uji	Penambahan data induk mahasiswa dengan data yang baru		
Tujuan	Memeriksa apakah data yang ditambahkan terrekam ke tabel induk mahasiswa		
Kondisi Awal	<input type="checkbox"/> Tabel induk mahasiswa sudah ada <input type="checkbox"/> Pemakai (administrator) sudah memilih (membuka) halaman web untuk penambahan data induk mahasiswa		
Tanggal Pengujian	05/01/2004		
Penguji	Kumkum		
Skenario			
1. Ketikkan data yang akan direkam (ditambahkan) 2. Klik tombol Simpan jika data sudah selesai diketik secara lengkap dan benar			
Hasil			
Data yang Diberikan	Yang Diharapkan	Pengamatan	Kesimpulan
NIM=113840178, Nama=Abdul Murudul, Jenis kelamin=L, Tempatlahir=Bandung, Tanggal lahir=17/08/65, Alamat=Jl. Dago 1000, Kota=Bandung, Kode pos=40123, No. telepon=2500052	Data mahasiswa terrekam ke tabel mahasiswa	<input type="checkbox"/> Field lain tidak dapat diisi jika NIM belum diisi <input type="checkbox"/> Tombol Simpan dapat di-klik jika field nama, jenis kelamin, tempat, tanggal lahir, alamat, dan kota sudah terisi <input type="checkbox"/> Pada database grid data baru ditempatkan di record terakhir	OK
Catatan			

Tugas Minggu Depan

1. Buatlah kelompok masing-masing terdiri dari 4 orang anggota.
2. Masing-masing kelompok menentukan sebuah software yang akan diuji
3. Software harus ada saat presentasi, tidak harus membuat dari awal (gunakan software yang sudah ada).
4. Buatlah laporan hasil pengujian secara Black Box (*Contoh Format Lampiran terlampir/ download di siadin*)
5. Presentasi kelompok
 - **A11.4803 – Jum'at, 21 November 2014**
 - **A11.4804 – Kamis, 20 November 2014**
6. **Semua anggota wajib presentasi, jika tidak nilai 0** (pengecualian untuk yang KKL Jakarta-Bandung bisa presentasi *Face to face* minggu berikutnya di ruangan).

TERIMA KASIH